REMARKS

Claims 1-8 and 10-21 are pending. Claims 1, 12 and 21 are independent.

The examiner maintained his rejection of claims 1 and 21 under 35 U.S.C. §101 on the ground that the claimed invention is directed to non-statutory subject matter.

Specifically, the examiner stated:

> 3. As to claims 1,21, Claims 1, 21 are not limited to tangible embodiments. In view of applicant's disclosure, specification page (2), line (3-15), the method or the medium is not limited to tangible embodiments, instead being defined as including both tangible embodiments (e.g., (hardware based processor)) and intangible embodiments (e.g., (internet). Although the claim recite the instructions cause the access of absolutely address and relatively address, based on broadest interpretation, it is read as intended use , not a positive limitation. The focus is not on the steps or feature taken to achieve the final result which is useful, tangible, and concrete, but rather the final result achieved which is useful, tangible, and concrete (see page 20, 101 Interim Guidelines published at uspto.gov). No final result which is useful, tangible, and concrete can be found in the claims. Therefore, it is directed non-statutory subject matter. (Office Action, page 2)

Applicant's independent claim 1 is performed in a parallel multithreaded processor, which is a tangible device. That parallel multithreaded processor includes a plurality of windows of registers, all of which are tangible, in which registers in those windows are relatively addressable by a corresponding thread, and are also absolutely addressable by two or more threads. Moreover, the tangible multithreaded processor recited in applicant's claim 1 is one in which the absolutely addressable feature is performed by providing the register's exact address in an instruction.

Accordingly, applicant's claim 1 recites a concrete, useful and tangible action, which is all that is necessary to make a claim statutory. Therefore, applicant's independent claim 1 recites statutory subject matter.

Applicant's independent claim 21 recites a "computer readable storage medium" which is a tangible element. Additionally, the instructions stored on the storage medium are such that they cause a multi-threaded processor to perform operations that, as was explained with respect to independent claim 1, result in a concrete, useful and tangible actions. Applicant's independent claim 21, therefore, recites statutory subject matter.

The examiner rejected claims 1-8 and 10-21 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,870,597 to Panwar et al. in view of U.S. Patent No. 5,996,068 to Dwyer. The examiner also rejected claims 1-6, 8, 10-18, 20 and 21 under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 5,900,025 to Sollars. Additionally, the examiner rejected claims 7, 19 under 35 U.S.C. §103(a) as being unpatentable over Sollars in view of Panwar.

With respect to the examiner's rejection of the claim 1 as being anticipated by Sollars, the examiner stated:

> 35. As to the newly amended claims 1,5,12,20, 21, Sollars disclosed maintaining execution thread in a parallel multithreaded processor (see the concurrent execution of eight threads in col.2, lines 1-14) comprising: a) accessing, by an executing thread in the multithreaded processor, a register set organized into a plurality of relatively addressable windows of registers that are relatively addressable per thread (see the virtually/physically addressable operand register sets in col.5, lines 20-31, see also the partitioned control register subsets in col.5, lines 32-54, col.6, lines 36-67, col.7, lines 1-7, col.7, lines 38-67, col.8, lines 1-11, see also fig.3 and fig.4 for corresponding virtual/physical addressable registers sets, see how register file [register set] organized into register sets [102][104][106] [register windows], see also how the register set 102 further organized into subsets 108). wherein accessing absolutely any one of the relatively (see virtually addressable) and absolutely addressable (see physically addressable) registers comprised providing an exact address of the register (see register addresses in col.8, line5-11), the exact address specified in an instruction associated with the thread (see the operand register sets accessible by the instruction in col.5, lines 25-26, see also the control registers accessible by instruction in col.5, lines 48-51).
> 36. As to the feature of register absolutely addressable by one or more threads, Sollars taught accessing absolutely any one of the relatively (see virtually addressable) and absolutely addressable (see physically addressable) registers comprised providing an exact address of the register by one or more threads (see register addresses in col.8, lines 5-11, see the concurrent execution of eight threads in col.2, lines 1-14). (Office Action, pages 9-10)

Further, responding to the applicant's arguments presented in applicant Amendment in Reply to Action of September 5, 2006, the examiner stated:

> 7. As to b), Sollars taught access of the different windows (see the different number of the dynamically associated thread 106 in col.7, lines 25-37), and the concurrent execution of eight threads for each of the active contexts (see col.2, lines 1-14). Therefore, a context control register was accessible by more than one thread. (Office Action, page 2)

Applicant disagrees with the examiner's contentions.

Sollars describes a processor with a number of control registers logically organized in a hierarchical manner to control the system, context and threads executed by Sollars' processor (see, for example, the Abstract and col. 1, line 60 to col. 2, line 6.) Specifically, Sollars explains that "[p]rimary control register file 20a comprises a plurality of control registers for performing the conventional functions of storing control and status information of executing processes (col. 5, lines 32-35) and describes the register file as follows:

> **FIG. 2 illustrates a logical view of the control registers under the presently preferred embodiment of the present invention. As shown, control registers of primary control register file 20a are organized into registers sets 102, 104, and 106, which in turn are organized into a hierarchy having three control register levels, i.e. a system level, a context level, and a thread level. At the highest level is a set of control registers 102 for controlling overall system operation. At the second highest level are multiple sets of control registers 104 for controlling concurrent execution of processes in multiple peer contexts. At the third level are multiple sets of control registers 106 for controlling concurrent execution of multiple peer process threads of the concurrently executing contexts. (FIG. 2 and col. 6, lines 35-49)**

Sollars further explains:

> **In one implementation of the preferred embodiment, eight sets of context control registers are provided for concurrently supporting up to eight active contexts, and 64 sets of thread control registers are provided for concurrently supporting up to eight active threads for each of the active contexts. Each of the system and context control register sets comprises 32 control registers, whereas each thread control register set comprises 16 control registers. (col. 2, lines 7-14)**

Thus, contexts control register sets each support a corresponding context. Contrary to the examiner's contentions with each context accessing its own context control register set.

Similarly, different threads associated with a particular context access only their own respective allocated thread control register sets. Specifically, as Sollars explains in relation to register allocation:

> **As shown in FIG. 12a, similarly, upon receipt of a valid request to create a new thread, i.e. from a thread with the proper context privilege, step 228, the context checks to determine if all thread level control register sets 106 have been allocated, step 230. If all thread level control register sets 106 have been allocated, the context further determines if the execution priority of the "new" thread requested is higher than at least one of the allocated threads, step 232. If the relative priority determination is unfavorable, the**

> **context simply queues the new thread request in a thread request queue of processor 10, step 234. On the other hand, if the relative priority determination is favorable, the context deallocates and queues the lowest priority allocated thread, step 236.**
>
> **Upon determining that there is at least one free thread level control register set 106 at step 230, or creating one through step 236, the context allocates a free thread level control register set 106 to the "new" thread, step 238. The context then requests the operating system to cause IFU 12 to reprioritize allocation of its internal resources. (col. 12, line 59 to col. 13, line 11)**

Thus, a new thread is allocated a free register set (i.e., a register set that is not being currently used).

Therefore, contrary to the examiner's contentions, different contexts/threads do not access any register set other than the one they have been allocated. Indeed, because control registers that are assigned to a particular context (or thread) are used primarily to maintain control of that context (or thread), there would be no need for another context (or thread) to access those control registers.

Accordingly, Sollars fails to disclose or suggest at least the feature of "each register in the plurality of windows of registers being relatively addressable by the corresponding thread and absolutely addressable by two or more of the threads executing on the multithreaded processor," as required by applicant's independent claim 1.

With respect to the examiner's rejections of claim 1 based on Panwar and Dwyer, the examiner admitted that "Panwar did not specifically show that the register was addressable by two or more threads executing on the multiprocessor as clamed." The examiner, however, stated:

> Dwyer taught a system for accessing a register set by a plurality of threads (see the sharing of the register pool for renaming the architectural registers of several threads in col.2 , lines 46-57). It would have been obvious to one of ordinary skill in the art to use Dwyer in Panwar for including the register addressable by two or more threads executing on the multiprocessor as clamed because the use of Dwyer could provide Panwar the ability to accept data from more than one thread process, and thereby increasing the ability to adapt to multiple bandwidth of maltreated tasks, and it could be achieved by predefining the multiple threads of Dwyer into the configuration file of Panwar with modified control parameters (e.g. the register map with the corresponding threads) so that multiple threads of Dwyer could be recognized by the register of Panwar, and because Panwar also taught a parallel multithreaded processor (see consistent implementation of multithreaded processor 102 in col.6, lines 40-44),which was a

> suggestion of the need for providing an access to a register, or the like, by a multiplicity of threads, and in doing so, provided a motivation. (Office Action, page 5, paragraph 13)

Applicant disagrees with the examiner's contentions.

Dwyer describes an apparatus and scheme for register renaming (col. 2, lines 7-9). Dwyer explains:

> **Another advantage gained with register renaming is the ability to overbook physical register assignment in a multi-threaded architecture. Many instruction set threads consist of only a fairly short number of instructions and only access a limited subset of the full architectural register set before they terminate, while some threads access the entire register set. Therefore, the average number of registers required by each thread is less than the full architectural set of registers. By sharing a pool of physical registers for renaming the architectural registers of several threads, a smaller physical register file can appear to provide full register sets to multiple threads. (col. 2, lines 46-57)**

Thus, according to Dwyer, multiple threads use only a limited number of registers available from a processor's register set, and therefore Dwyer's apparatus maps registers identified by particular threads to other registers of the register set so that a greater utilization of the available registers may be achieved. Accordingly, contrary to the examiner's contentions, Dwyer's apparatus results in different registers of a register set being accessed by different threads. Dwyer, therefore, fails to disclose or suggest "each register in the plurality of windows of registers being relatively addressable by the corresponding thread and absolutely addressable by two or more of the threads executing on the multithreaded processor," as required by applicant's independent claim 1.

Because none of Sollars, Panwar and Dwyer discloses or suggests, alone or in combination, at least the feature of "each register in the plurality of windows of registers being relatively addressable by the corresponding thread and absolutely addressable by two or more of the threads executing on the multithreaded processor," applicant's independent claim 1, and the claims that depend from it are therefore patentable over the cited art.

Independent claims 12 and 21 recite "each register in the plurality of windows of registers being relatively addressable by the corresponding thread associated with the respective window of registers and absolutely addressable by two or more of the threads executing on the multi-threaded processor," or similar language. For reasons similar to those provided with

respect to independent claim 1, at least this feature is not disclosed by the cited art. Accordingly, independent claims 12 and 21, and the claims that depend from them, are patentable over the cited art.

It is believed that all the rejections and/or objections raised by the examiner have been addressed.

Canceled claims, if any, have been canceled without prejudice or disclaimer.

Any circumstance in which the applicant has (a) addressed certain comments of the examiner does not mean that the applicant concedes other comments of the examiner, (b) made arguments for the patentability of some claims does not mean that there are not other good reasons for patentability of those claims and other claims, or (c) amended or canceled a claim does not mean that the applicant concedes any of the examiner's positions with respect to that claim or other claims.

In view of the foregoing remarks, applicant respectfully submits that the application is in condition for allowance and such action is respectfully requested at the examiner's earliest convenience.

No fee is believed due. Please apply any required charges to deposit account 06-1050, referencing the attorney docket number shown above.

Respectfully submitted,

Date: April 20, 2007

Ido Rabinovitch
Attorney for Intel Corporation
Reg. No. L0080

PTO Customer No. 20985
Fish & Richardson P.C.
Telephone: (617) 542-5070
Facsimile: (617) 542-8906